1.a

50

20 60

10 40 70

15 65 80

30 72

25 38

1.b

Pre-order: 50 20 10 15 40 30 25 38 60 70 65 80 72

In-order: 10 15 20 25 30 38 40 50 60 65 70 72 80

Post-order: 15 10 25 38 30 40 20 65 72 80 70 60 50

1.c after deleting 30 then 20

50

25 60

10 40 70

15 65 80

38 72

2 a.)

struct BSTNode

{

Node(int value, BSTNode\* parent)

: m\_value(value), l\_child(nullptr), r\_child(nullptr), m\_parent(parent)

{}

int m\_value;

BSTNode\* l\_child, \*r\_child, \* m\_parent;

};

3.b.)

void insert(BSTNode\* curr, int value, BSTNode\* parent)

{

insertHelper(curr, value, nullptr);

}

void insertHelper(BSTNode\* curr, int value, BSTNode\* parent)

{

if(curr is nullptr)

create a new BSTNode with the data as the value, the parent pointer being set to parent

and the left and right children to be set as nullptr

else if(curr's value is less than the value passed in)

insertHelper(curr's left child, value, curr)

else if(curr's value is greater than the value passed in)

insertHelper(curr's right child, value, curr)

}

3.a

7

3 6

0 2 4

3.b

Array representation: 7 3 6 0 2 4

3.c

Array representation: 6 3 4 0 2

4.

a. O(C+ S)

b. O(log(C) +S)

c. O(log(C) + log(S))

d. O(1+log(S))= O(log(S))

e. O(1+1)= O(1)

f. O(log(C) + S)

g. O(S\* log(S))

h. O(C\*log(S)